CLAIMS

1. A method of producing obfuscated object code, the method comprising the steps of substituting a variable in source code with a selected function of the variable, and compiling the source code to produce object code, the selected function causing the variable to be presented in the compiled object code as a series of operations.

2. A method as claimed in Claim 1, wherein the series of operations by which the variable is presented is made up of arithmetic and/or logical operations, and wherein the series of operations is arranged, upon running of the object code, to provide the variable.

3. A method as claimed in Claim 1 or Claim 2, wherein the series of operations by which the variable is presented comprises complementary operations arranged, upon running of the object code, to provide the variable.

4. A method as claimed in any preceding claim, wherein the selected function is defined in a template of the source code.

5. A method as claimed in Claim 4, wherein the template of the source code defines a plurality of functions which are each arranged to compile to present the variable as a series of operations, the method further comprising selecting one of the functions to substitute for the variable in the source code.

6. A method as claimed in Claim 5, wherein a different key is associated with each one of the functions in the template, and the method further comprises substituting the variable in source code with the template and selecting one of the functions in the template by selecting the key which is associated with said one function.

7. A method as claimed in any preceding claim, wherein the source code involves stored arrays and templates and utilises pointers to navigate the arrays and templates.

8. A method as claimed in any preceding claim, wherein the source code is

a standard programming language.

9. A method as claimed in Claim 8, wherein the source code is C^{++}.

10. An executable program in object code, the program having been compiled from source code, wherein a variable in the source code has been compiled to be presented in object code as a series of operations whereby the object code is obfuscated.

11. An executable program as claimed in Claim 10, wherein the series of operations by which the variable is presented is made up of arithmetic and/or logical operations, and wherein the series of operations is arranged, upon running of the object code, to provide the variable.

12. An executable program as claimed in Claim 10 or Claim 11, wherein the series of operations by which the variable is presented comprises complementary operations arranged, upon running of the object code, to provide the variable.

13. An executable program as claimed in any of Claims 10 to 12, wherein the series of operations has been produced by substituting the variable in the source code with a selected function arranged to cause the variable to be presented in the compiled object code as a series of operations.

14. An executable program as claimed in Claim 13, wherein the selected function was defined in a template of the source code.

15. An executable program as claimed in Claim 14, wherein the template of the source code had defined a plurality of functions which were each arranged to compile to present the variable as a series of operations, and one of the functions had been selected to substitute for the variable in the source code.

16. A method of producing storage media having a secured executable program thereon, the method comprising the steps of securing an executable program by associating the executable program with a security program which is arranged to control access to the executable program, and applying the

secured executable program to the storage media, and the method further comprising obfuscating the object code of the security program, wherein the object code of the security program has been obfuscated by substituting a variable in source code with a selected function of the variable, and compiling the source code to produce object code, the selected function causing the variable to be presented in the compiled object code as a series of operations.

17. A method of producing storage media having a secured executable program thereon as claimed in Claim 16, wherein the series of operations by which the variable is presented is made up of arithmetic and/or logical operations, and wherein the series of operations is arranged, upon running of the object code, to provide the variable.

18. . A method of producing storage media having a secured executable program thereon as claimed in Claim 16 or Claim 17, wherein the series of operations by which the variable is presented comprises complementary operations arranged, upon running of the object code, to provide the variable.

19. A method of producing storage media having a secured executable program thereon as claimed in any of Claims 16 to 18, wherein the executable program and the security program are associated at object code level.

20. A method of producing storage media having a secured executable program thereon as claimed in any of Claims 16 to 19, wherein the security program is arranged to encrypt the executable program.

21. A method of producing storage media having a secured executable program thereon as claimed in any of Claims 16 to 20, further comprising moving blocks of the executable program out of the executable program and relocating the blocks in the security program.

22. A method of producing storage media having a secured executable program thereon as claimed in any of Claims 16 to 21, wherein the security program is arranged to require the running of an authentication program.

23. A method of producing storage media having a secured executable

program thereon as claimed in any of Claims 16 to 22, wherein the selected function in the source code of the security program is defined in a template of the source code.

24. A method of producing storage media having a secured executable program thereon as claimed in Claim 23, wherein said template of the security program source code defines a plurality of functions which are arranged to compile to present the variable as a series of operations, the method further comprising selecting one of the functions to substitute for the variable in the source code.

25. A method of producing storage media having a secured executable program thereon as claimed in Claim 24, wherein a different key is associated with each one of the functions in the template, and the method further comprises substituting the variable in source code with the template and selecting one of the functions in the template by selecting the key which is associated with said one function.

26. A method of producing storage media having a secured executable program thereon as claimed in any of Claims 16 to 25, wherein the source code of the security program involves stored arrays and templates and utilises pointers to navigate the arrays and templates.

27. A method of producing storage media having a secured executable program thereon as claimed in any of Claims 16 to 26, wherein the source code of the security program is a standard programming language.

28. A method of producing storage media having a secured executable program thereon as claimed in Claim 27, wherein the source code is C^{++}.

29. A method of producing storage media having a secured executable program thereon as claimed in any of Claims 16 to 28, wherein the storage media onto which the secured executable program is applied is an optical disc.

30. A method of producing storage media having a secured executable program thereon as claimed in Claim 29, wherein the secured executable

program is applied onto the optical disc by laser beam encoding.

31. A method of producing storage media having a secured executable program thereon as claimed in any of Claims 16 to 28, wherein the storage media onto which the secured executable program is applied is memory in, or associated with, servers, computers and/or other processing means.

32. A storage media having a secured executable program thereon, wherein an executable program is secured by having a security program associated therewith, the security program being arranged to control access to the executable program, and wherein the security program is in object code which has been obfuscated, the security program having been compiled from source code, and a variable in the source code of the security program having been compiled to be presented in object code as a series of operations whereby the object code has been obfuscated.

33. A storage media having a secured executable program thereon as claimed in Claim 32, wherein the series of operations by which the variable is presented is made up of arithmetic and/or logical operations, and wherein the series of operations is arranged, upon running of the object code, to provide the variable.

34. A storage media having a secured executable program thereon as claimed in Claim 32 or Claim 33, wherein the series of operations by which the variable is presented comprises complementary operations arranged, upon running of the object code, to provide the variable.

35. A storage media having a secured executable program thereon as claimed in any of Claims 32 to 34, wherein the series of operations has been produced by substituting the variable in the source code of the security program with a selected function arranged to cause the variable to be presented in the compiled object code as a series of operations.

36. A storage media having a secured executable program thereon as claimed in Claim 35, wherein the selected function was defined in a template of the source code.

37. A storage media having a secured executable program thereon as claimed in Claim 36, wherein the template of the source code had defined a plurality of functions which were each arranged to compile to present the variable as a series of operations, and one of the functions had been selected to substitute for the variable in the source code.

38. A storage media having a secured executable program thereon as claimed in any of Claims 32 to 37, wherein the executable program and the security program are associated at object code level.

39. A storage media having a secured executable program thereon as claimed in any of Claims 32 to 38, wherein the executable program is encrypted on the storage media and the associated security program enables decryption of the executable program.

40. A storage media having a secured executable program thereon as claimed in any of Claims 32 to 39, wherein blocks from the executable program have been relocated within the security program.

41. A storage media having a secured executable program thereon as claimed in any of Claims 32 to 40, wherein the security program is arranged to require the running of an authentication program.

42. A storage media having a secured executable program thereon as claimed in any of Claims 32 to 41, wherein the storage media is an optical disc on which the executable program and the security program are encoded.

43. A storage media having a secured executable program thereon as claimed in Claim 42, wherein the optical disc is a CD, a CD-ROM, or a DVD.

44. A storage media having a secured executable program thereon as claimed in any of Claims 32 to 41, wherein the storage media is memory in, or associated with, servers, computers and/or processing means and on which the executable program and the security program are stored.

45. A storage media having a secured executable program thereon as claimed in any of Claims 32 to 44, wherein the executable program is a games program, and/or a video program, and/or an audio program, and/or other software.

46. A method of controlling a processor to run a program comprising the steps of:

translating instructions from the program into a reduced instruction set format to which said processor is not responsive,

causing the translated instructions to be applied to a virtual processor which is responsive to the reduced instruction set format, and

causing the virtual processor to run the instructions applied thereto and to apply a series of simple instructions, to which the processor is responsive, to the processor.

47. A method of controlling a processor to run a program as claimed in Claim 46, further comprising encrypting the translated instructions to be applied to the virtual processor, and enabling the virtual processor to respond to the encrypted instructions without decrypting them.

48. A method of controlling a processor to run a program as claimed in Claim 46 or Claim 47, further comprising utilising templates to translate and encrypt the instructions, a selected template providing a series of instructions in the reduced instruction set format for each instruction from the program.

49. A method of controlling a processor to run a program as claimed in Claim 48, wherein the templates define a plurality of series of instructions in the reduced instruction set format for an instruction in the program, the method further comprising selecting one of said plurality of series of instructions to be the translation for said instruction.

50. A method of controlling a processor to run a program as claimed in Claim 49, wherein a different key is associated with each one of the plurality of series of instructions in the reduced instruction set format in the template, and wherein the method further comprises selecting a key which is associated with one of said plurality of series of instructions and translating the instruction in the

program to the one of said plurality of series of instructions which is associated with the selected key.

51. A method of controlling a processor to run a program as claimed in Claim 50, wherein the virtual processor is enabled to respond to said one of said plurality of series of instructions.

52. A method of controlling a processor to run a program as claimed in any of Claims 46 to 51, wherein only instructions from the program which perform predetermined functions or routines are translated and applied to said virtual processor.

53. A method of controlling a processor to run a program as claimed in any of Claims 46 to 51, wherein the instructions from the program are in object code or have been transformed from object code, and wherein the object code has been obfuscated by a method as claimed in any of Claims 1 to 9.

54. A method of controlling a processor to run a program as claimed in any of Claims 46 to 53, wherein the instructions are from an executable program in object code as claimed in any of Claims 10 to 15.

55. A method of controlling a processor to run a program as claimed in any of Claims 46 to 52, wherein the instructions are from a secured executable program on a storage media produced by a method as claimed in any of Claims 16 to 31.

56. A method of controlling a processor to run a program as claimed in any of Claims 46 to 54, wherein the instructions are from a secured executable program on a storage media as claimed in any of Claims 32 to 45.

57. A storage media having a secured executable program thereon, wherein an executable program is secured by having a security program and an emulation program associated therewith, the security program being arranged to control access to the executable program, and the emulation program causing predetermined functions or routines of the executable program to be run on a virtual processor provided by said emulation program, wherein the

emulation program is arranged to translate instructions from the executable program into a reduced instruction set format, to cause the translated instructions to be applied to the virtual processor, and to cause the virtual processor to run the instructions applied thereto and to output a series of simple instructions for application to a processor.

58. A storage media having a secured executable program thereon as claimed in Claim 57, wherein the storage media is an optical disc on which the executable program, the security program and the emulation program are encoded.

59. A storage media having a secured executable program thereon as claimed in Claim 57 or Claim 58, wherein the optical disc is a CD, a CD-ROM or a DVD.

60. A storage media having a secured executable program thereon as claimed in Claim 57, wherein the storage media is memory in, or associated with, servers, computers and/or other processing means and on which the security program and the emulation program are stored.

61. A storage media having a secured executable program thereon as claimed in any of Claims 57 to 60, wherein the executable program is a games program and/or a video program and/or an audio program, and/or other software.

62. A method of producing obfuscated object code substantially as hereinbefore described with reference to the accompanying drawings.

63. An executable program in object code, substantially as hereinbefore described with reference to the accompanying drawings.

64. A method of producing storage media having a secured executable program thereon substantially as hereinbefore described with reference to the accompanying drawings.

65. A storage media having a secured executable program thereon

substantially as hereinbefore described with reference to the accompanying drawings.

66. A method of controlling a processor to run a program substantially as hereinbefore described with reference to the accompanying drawings.